Remarks

Entry of this amendment, reconsideration of the application and allowance of all claims are respectfully requested. Claims 1, 3-9, 12, 14-20, 23-25, 27-33 & 36-39 are now pending.

By this paper, independent claims 1, 12, 23 & 25 are amended and new claims 36-39 are added to more particularly point out and distinctly claim the subject matter of the present invention. Moreover, Applicants have canceled claims 2, 10, 11, 13, 21, 22, 26, 34 & 35 without prejudice, with the subject matter recited therein now being incorporated into the respective independent claims. These amendments to the claims presented herein constitute a *bona fide* attempt by the Applicants to advance prosecution of this application and obtain allowance of certain claims and are in no way meant to acquiesce to the substance of the Examiner's rejections. It is believed that the amendments to the claims place all claims in condition for allowance. Support for the amended claims can be found throughout the application as filed. For example, reference canceled claims 2, 10, 11, 13, 21, 22, 26, 34 & 35, as well as paragraphs 37-39, 42, 54, 55, 67-70 and 72-85 of the application. No new matter is added to the application by any amendment presented.

Substantively, original claims 1-4, 8, 10-15, 19, 21-28, 32 & 34-35 were rejected under 35 U.S.C. §102(a) as being anticipated by Darty (U.S. Patent No. 6,173,440), while claims 5-7, 9, 16-18, 20, 29-31 & 33 were rejected under 35 U.S.C. §103(a) as being unpatentable over Darty. These rejections are respectfully traversed to any extent deemed applicable to the amended claims presented herewith.

In one aspect, Applicants' invention is directed to a technique for evaluating a software component (e.g., claims 1, 12, 23 & 25). This technique includes deriving a conceptual layering scheme of the software component. The conceptual layering scheme divides the software component into multiple software layers and the multiple layers take into account relationships that exist between the software layers. The technique further includes providing an abstraction matrix in mathematical abstract form that describes the multiple layers of the software component and accounts for the relationships that exist between the software layers. This abstraction matrix further includes at least one test case scenario for each software layer and mapped expected results therefor. The technique also comprises testing the multiple software

POU92000018311S1   -10-

layers of the software component using the test case scenarios of the abstraction matrix to generate test results, and evaluating the test results using the abstraction matrix. The evaluating includes for each software layer automatically comparing a test case employed in the testing of the layer to the at least one test case scenario of the abstraction matrix therefor, and if a match is found, automatically comparing the test results from the test case with the mapped expected results for that test case scenario in the abstraction matrix. Evaluation of the software component is accomplished upon completion of the evaluation of the test results. Advantageously, Applicants' deriving of a conceptual layering scheme for the software component and accounting for relationships that exist between software layers allows independent test per software layer to be constructed, thus significantly reducing the number of tests needed to verify the software component.

With respect to the anticipation rejection, it is well settled that there is no anticipation of a claim unless a single prior art reference discloses: (1) all the same elements of the claimed invention; (2) found in the same situation as the claimed invention; (3) united in the same way as the claimed invention; and (4) in order to perform the identical function as the claimed invention. In this instance, Darty fails to disclose various aspect of Applicants' invention as recited in the independent claims presented, and as a result, does not anticipate (or even render obvious) Applicants' invention.

Darty discloses an approach for software debugging, verification and validation. A knowledge-based reasoning approach is employed to build a functional model of the software code for identifying and isolating failures of the software code. This knowledge-based reasoning approach uses the software design, which is preferably based upon a flow chart or block diagram representation of the software functionality, to build the functional model. The software block diagram contributes to the functional model by defining the inputs and outputs of the various blocks of code, as well as defining data interconnections between the various blocks of code. In accordance with a method of the present invention, test points are strategically inserted throughout the code, and each test point is associated with a corresponding block of code. Expected values of the test points for an expected proper-operation execution of the computer program are generated. The computer program is then executed on a computer, and the actual values of the test points from the program execution are compared with the expected

values of the test points. Failed test points which do not agree with corresponding expected values are determined. The functional model, which includes information functionally relating the various test points to one another, is then used to isolate the failed test points to one or more sources of failure in the code. (See Abstract.)

Initially, Applicants respectfully submit that Darty does not teach or suggest their technique of software component evaluation which includes deriving a conceptual layering scheme for the software component, let alone a conceptual layering scheme that divides the software component into multiple software layers, with the multiple software layers taking into account relationships that exist between the layers. To the extent relevant to this concept, the Office Action states at paragraph 9, pages 6 & 7:

> As per claims 10, 21, and 34, Darty discloses that testing of the software component is based on layer of the software component, and wherein the evaluating comprises evaluating the test results for at least one layer of the software component using the abstraction matrix (abstract; Figure 1 – elements 20, 22, 24, and associated text; Figure 6 – "commercial diagnostic analyss tool: (emphasis), "capture design in CAD", "analysis", "select test point locations", and associated text; Figure 7 and associated text; Figure 8 and associated text (emphasis added to "select software", "test points", "analyze testability of software design", "does the design meet testability requirement", and "redesign software to correct failures"); Figure 9 and associated text (emphasis added to "write software code to determine pass/fail for each test point in each operational mode", "run software and determine pass/fail data for each test point", "are there any failures of any test points?", "execute diagnostics for fault isolation using knowledge base as mode", and "diagnostic results with isolated faults"); Figure 10 – "execute software and real time diagnostics", "process failure data and determine corrective action", "software verified and validated", "establish test points and perform design analysis for testability", "generate knowledge base of design", and associated text; col. 2:13 to col. 3:48 (emphasis added to functional model, software block diagram, test points, blocks of code, execution of the software code, expected values, actual values, comparison of actual values of the test points to the expected values of the test points)).

Applicants respectfully submit that the above-noted citations of various words and phrases from Darty does not establish a teaching or a suggestion for one of ordinary skill in the art of Applicants' recited 'deriving a conceptual layer scheme for the software component'. In fact, a careful reading of Darty fails to uncover any discussion of Applicants' recited concept of

POU920000183US1   -12-

deriving a conceptual layering scheme which divides the software component into multiple software layers, let alone Applicants' approach wherein the layers take into account relationships that exist between the layers. There is simply no teaching in Darty of deriving a conceptual layering scheme *per se* for the software component under evaluation. Applicants note that the word "layer" does not even appear in the above-cited material from the Darty patent. Should the Examiner believe otherwise, Applicants request that the Examiner explain the applicability of Darty to their recited process for deriving a conceptual layering scheme of the software component, and particularly, a layering scheme wherein the software component is divided into multiple software layers and the multiple layers take into account relationships that exist between the layers. This concept is believed unique to Applicants' invention.

Applicants further recite providing an abstraction matrix in mathematical abstract form that describes the multiple software layers of the software component and accounts for the relationships that exist between the layers. This abstraction matrix further includes at least one test case scenario for each layer and mapped expected results therefor. A careful reading of Darty fails to uncover any similar teaching or suggestion. Darty describes generation of expected values of test points for an expected proper-operation execution of a computer program, but does not teach or suggest providing an abstraction matrix that describes the multiple software layers of the software component and accounts for the relationships between the layers, nor an abstraction matrix that has for each layer at least one test case scenario and mapped expected results therefor. Paragraph 10 at page 7 of the Office Action discussed the subject matter of original claims 11, 22 & 35 and states:

> As per claims 11, 22 & 35. Darty discloses that the providing comprises providing the abstraction matrix to include at least one test case scenario for each layer of the software component (abstract; Figure 1 – elements 20, 22, 24, and associated text; Figure 6 - "commercial diagnostic analysis tool" (emphasis), "capture design in CAD", analysis". "select test point locations", and associated text; Figure 7 and associated text; Figure 8 and associated text; Figure 10 - "establish test points and perform design analysis for testability", "generate knowledge base of design", and associated text; col. 2:13 to col. 3:48 (emphasis added to functional model, software block diagram, test points, actual values, expected values)).

Applicants respectfully submit that the above-noted figures and quoted phrases from Darty do not describe the functionality recited by Applicants in connection with their abstraction matrix of the independent claims. Again, Darty does not disclose the concept of layering *per se*, let alone providing an abstraction matrix of the software component that accounts for relationships that exist between the software layers, nor the provision in the abstraction matrix of a test case scenario for each software layer of the software component and mapped expected results therefor.

Still further, Applicants recite in their amended independent claims functionality which includes testing the multiple layers of a software component using the test case scenarios in the abstraction matrix to generate test results, and evaluating the test results using the abstraction matrix. This evaluating includes for each layer automatically comparing a test case employed in the testing of the layer to the at least one test case scenario of the abstraction matrix therefor, and if a match is found, automatically comparing the test results for that test case with the match expected results for that test case scenario in the abstraction matrix. Evaluation of the software component is accomplished upon completion of evaluation of the test results. Applicants respectfully submit that the cited words and phrases and figures from Darty at pages 2-4 of the Office Action do not teach or suggest their cited functionality in the amended independent claims.

To summarize, Applicants respectfully submit that Darty fails to teach or suggest various aspects of their recited evaluation process, including: (i) deriving a conceptual layering scheme of the software component, the conceptual layering scheme dividing the software component into multiple software layers, the multiple layers taking into account relationships that exist between the software layers; (ii) providing an abstraction matrix in mathematical abstract form that describes the multiple layers of the software component and accounts for the relationships that exist between the software layers, the abstraction matrix further comprising at least one test case scenario for each layer and mapped expected results therefor; and (iii) testing the multiple layers of the software component using the test case scenarios to generate test results, and evaluating the test results using the abstraction matrix as recited in their independent claims. Further, there is no teaching or suggestion in the cited art that would lead one of ordinary skill in

POU920000183US1   -14-

the art to modify Darty to somehow achieve an evaluation process as recited by Applicants' in the claims presented.

For all of the above reasons, Applicants respectfully submit that the amended independent claims 1, 12, 23 & 25 patentably distinguish over the teachings of Darty. Reconsideration and withdrawal of the anticipation rejection based thereon is therefore respectfully requested.

With respect to the obviousness rejection to various dependent claims, Applicants note than an "obviousness" determination requires an evaluation of whether the prior art taken as a whole would suggest the claimed invention taken as a whole to one of ordinary skill in the art. In evaluating claimed subject matter as a whole, the Federal Circuit has expressly mandated that functional claim language be considered in evaluating a claim relative to the prior art. Applicants respectfully submit that the application of these standards to the amended claims at issue leads to the conclusion that the recited subject matter would not have been obvious to one of ordinary skill in the art based on the applied patents.

Again, a careful reading of Darty fails to uncover any teaching or suggestion of the above-noted deficiencies thereof when applied against the independent claims presented. Since Darty does not disclose Applicants' recited concept of deriving a conceptual layering scheme for the software component to be evaluated, wherein the conceptual layering scheme divides the software component into multiple software layers, and the multiple layers take into account relationships that exist between the layers. Applicants respectfully submit that all of the claims presented patentably distinguish over the applied art, including new claims 36-39 (consideration of which is respectfully requested).

Thus, the application is believed to be in condition for allowance and such action is respectfully requested.

Applicants' undersigned attorney is available should the Examiner wish to discuss this application further.

Respectfully submitted.

Kevin P. Radigan
Attorney for Applicants
Registration No.: 31,789

Dated: April 04, 2005.

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 Columbia Circle
Albany, New York 12203-5160
Telephone: (518) 452-5600
Facsimile: (518) 452-5579

POU920000183US1   -16-